User's Guide for the NMM Core of the Weather Research and Forecast (WRF) Modeling System Version 2.1

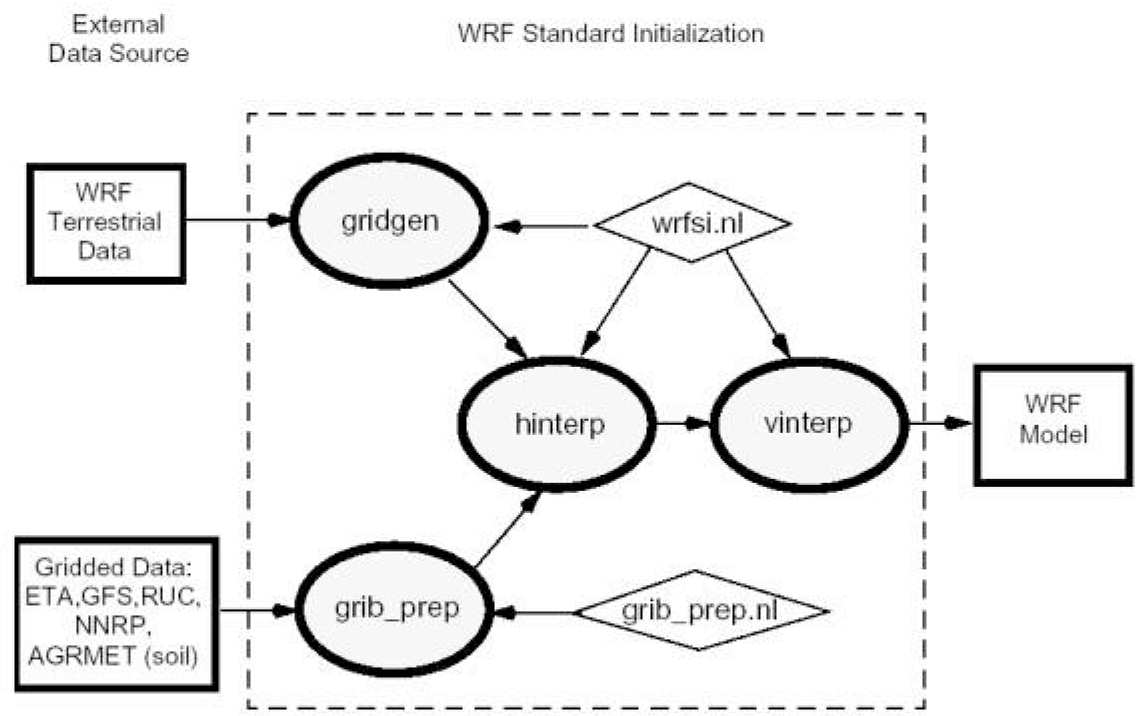
Chapter 3: WRF-NMM Standard Initialization (WRF-NMM SI) Preparing Input Data

Table of Contents

Introduction

The WRF-NMM Standard Initialization (SI) is the first step in the process of producing a WRF-NMM real-data simulation or forecast. The software is a collection of four programs that together provide the input data required by the WRF-NMM model. Although the ARW and NMM dynamic cores within the WRF framework currently have separate SI packages, both packages have the same basic components that perform the same functions. The following figure illustrates the program components and data flow for both SI packages:

WRFSI Flow Chart (*for WRFV2*)



The SI package provides the tools for defining a mesoscale domain. It ingests data from various terrestrial datasets for terrain, land use, soil type, annual deep soil temperature, monthly vegetation fraction, maximum snow albedo, monthly albedo, slope data, and meteorological data from another model (in GriB format). It interpolates the data to the user defined domain. The output from SI is in netCDF format and conforms to WRF I/O API.

The WRF-NMM SI has been successfully ported to UNIX based machines (IBM) and LINUX (using PGI compiler).

The WRF-NMM SI code runs on single processor machines only. The code is memory efficient.

Function of WRF-NMM SI Program

The SI package consists of four major independent programs: ***grib_prep***, ***gridgen_model***, ***hinterp*** and ***vinterp***. The package also includes two utility programs, ***siscan*** and ***plotfmt***.

Program *gridgen_model*

The function of the program ***gridgen_model*** is to define a simulation domain, and read and interpolate various terrestrial datasets from latitude/longitude grid to the projection grid. The simulation domain is defined based on information specified by the user in the `'hgridspec'` section of the namelist file, ***wrfsi.nl***. The terrestrial inputs that ***gridgen_model*** uses include terrain, land use, soil type, annual deep soil temperature, monthly vegetation fraction, maximum snow albedo, monthly albedo, and slope data. These data are provided at: http://wrfsi.noaa.gov/release/

NOTE: The WRF-NMM SI only supports the "Rotated Latitude Longitude" projection.

Program *grib_prep*

The function of the program ***grib_prep*** is to read GriB files, degrib the data, and write the data out in a simple format, which is referred to as the intermediate format. The GriB files contain time-varying meteorological fields and are typically from another regional or global model, such as NCEP's NAM (formerly Eta) and GFS models. WRF-NMM SI supports the GriB format Edition 1.

Each GriB dataset may contain more data than is needed to initialize WRF-NMM. To limit the data ingested from the GriB files, Vtables are employed by which GriB and level codes are used to identify a particular field. Different GriB files may have different codes for the same variable; hence different Vtables are prepared for commonly available GriB files.

The Vtables provided are for NAM/Eta 221 and 212 grids, NAM/Eta data in AWIP format, GFS (or AVN), NCEP/NCAR Reanalysis archived at NCAR, RUC (pressure level data and hybrid coordinate data), GFDL, GFS, NNRP and AFWA's AGRMET land surface model output.

To use a GriB dataset for which a Vtable is not provided, follow the Vtable examples in the WRF-NMM SI tar file under the directory `extdata/static/` to create a Vtable for this dataset.

The intermediate format can also be used to ingest any dataset as long as the data are on pressure levels (data on other coordinates will require code modifications). A description of the intermediate format can be found at http://wrfsi.noaa.gov/, or in the README file in the WRF-NMM SI tar file.

Program hinterp

The function of the program ***hinterp*** is to horizontally interpolate meteorological data degribbed by ***grib_prep*** onto the simulation domain created by ***gridgen_model***. The methods of horizontal interpolation are controlled by variables in the `'interp_control'` section of ***wrfsi.nl***.

Program *vinterp*

The function of the program ***vinterp*** is to vertically interpolate meteorological data from pressure (or hybrid data in the case of RUC) levels to the WRF-NMM vertical coordinate defined by the user in the ***'interp_control'*** section of ***wrfsi.nl***.

Utility Program *siscan*

Program ***siscan*** is a utility program that may be used to read ***hinterp*** output (file name begins with hinterp.d01.*).

```
     siscan hinterp.d01.2005-04-27_00:00:00
```

```
Scanning hinterp.d01.2005-04-27_00:00:00

 Domain Metadata Information
 -----------------------------------------------------
Domain Number .........  1
Parent ID .............  0
Dynamic Init. Source .. SI
Static Init. Source ... SI
Valid Date (YYYDDD) ... 2005117
Valid Time (sec UTC) ..     0.0
Origin X in Parent ....     1
Origin Y in Parent ....     1
Nest Ratio to Parent ..     1
Delta X ...............  8341.6
Delta Y ...............  8341.6
Top Level .............   5000.0
Origin Z in Parent ....     1
X dimension ...........   123
Y dimension ...........   201
Z dimension ...........    40
  ---------------------------------------------------
Variables found:
   NAME    STAG DIM   NX    NY    NZ        UNITS        DESCRIPTION
 -------- ----- --- ----- ----- ----- ---------------- --------------------------------
PRESSURE     0   1    40     0     0 Pa               Pressure levels used for vertica
T            4   3   123   201    40 K                Temperature
U            4   3   123   201    40 m s{-1}          U
V            4   3   123   201    40 m s{-1}          V
RH           4   3   123   201    40 %                Relative Humidity
HGT          4   3   123   201    40 m                Height
PMSL         4   2   123   201     0 Pa               Sea-level Pressure
PSFC         4   2   123   201     0 Pa               Surface Pressure
WEASD        1   2   123   201     0 kg m{-2}         Water Equivalent of Accum Snow D
SKINTEMP     1   2   123   201     0 K                Sea-Surface Temperature
ST000010     1   2   123   201     0 K                T of 0-10 cm ground layer
ST010040     1   2   123   201     0 K                T of 10-40 cm ground layer
ST040100     1   2   123   201     0 K                T of 40-100 cm ground layer
ST100200     1   2   123   201     0 K                T of 100-200 cm ground layer
TGROUND      4   2   123   201     0 K                T at 300 cm deep in ground
SM000010     1   2   123   201     0 fraction         Soil Moisture of 0-10 cm ground
SM010040     1   2   123   201     0 fraction         Soil Moisture of 10-40 cm ground
SM040100     1   2   123   201     0 fraction         Soil Moisture of 40-100 cm groun
SM100200     1   2   123   201     0 fraction         Soil Moisture of 100-200 cm grou
SEAICE       1   2   123   201     0 0/1 Flag         Ice flag
SST          1   2   123   201     0 K                Sea Surface Temperature
CANWAT       1   2   123   201     0 kg m{-2}         Plant Canopy Surface Water
SPECHUMD     4   2   123   201     0 kg kg{-1}        Specific Humidity
SOILHGT      0   2   123   201     1 m                Terrain height of source data
End of file reached.
```

Utility Program *plotfmt*

Utility program ***plotfmt*** can be used to make simple graphics from the intermediate files. This program plots every field in the intermediate formatted data file. This utility is not automatically built when SI is installed. To compile this program, cd to the ***src/grib_prep/util*** directory and type:

```
make plotfmt.exe
```

To run this utility program, type

```
plotfmt intermediate-file-name
```

How to Install WRF-NMM SI

The WRF-NMM SI program may be downloaded from the http://wrfsi.noaa.gov/ page. An 'Installation README' file is posted on the site. A summary of the installation procedure is provided below.

Required Compilers, Scripting Language and Libraries

WRF-NMM SI code is written mostly in FORTRAN 77 and FORTRAN 90. A few utility programs are written in C. Hence, FORTRAN 90 and C compilers (gcc is recommended) are required,which are also required for the WRF model. Perl scripts are used to run the SI programs, and perl/Tk is required to run the GUI. WRF-NMM SI writes output in both binary (from *grib_prep* and *hinterp*) and netCDF (from *gridgen_model* and *vinterp*). As with the WRF model, a pre-installed netCDF library is required.

Hint: Using PGI or Intel compiler on a **LINUX** computer requires the netCDF library be installed using the same compiler.

Installation Steps

- Download the program tar file, type ***'gunzip wrfnmm_si_v2.1.tar.gz'*** to unzip the file, and ***'tar -xf wrfnmm_si_v2.1.tar'*** to untar the file. These steps will create a directory called ***wrfsi_nmm/,*** which will be referred to as the SOURCE_ROOT directory. This directory contains:

  HOW_TO_RUN.txt: instructions how to run WRF-NMM SI without using the GUI
  HOW_TO_INSTALL: instructions on how to install WRF-NMM SI
  Makefile: top-level makefile
  README_wrfnmm_si: WRF-NMM SI documentation
  README.wrfsi.nl: description of namelist variables
  data/: default directory for running the WRF-NMM SI, containing namelist file

etc/: directory where scripts for running SI programs reside
extdata/:default directory for degribbed data files
graphics/: directory where NCL scripts reside
gui/: source directory for the GUI
src/: the source code directory
util/: utilities for installation and IDL routines.

- Decide on a directory structure for the following

  - executables and Perl scripts that run various SI programs: INSTALLROOT
  - terrestrial datasets (terrain, land use, etc.): GEOG_DATAROOT
  - intermediate formatted data files: EXT_DATAROOT
  - top level directory for user defined domains: DATAROOT
  - directory for specific user defined domain: MOAD_DATAROOT
  - template directory: TEMPLATES

  Note: *DATAROOT can be a top directory that contains multiple MOAD_DATAROOT subdirectories or MOAD_DATAROOT can be the same as the DATAROOT if there is only one MOAD_DATAROOT*

- Set the following environment variables to reflect the directory structure above:

```
setenv SOURCE_ROOT source-root-directory
setenv INSTALLROOT install-root-directory
setenv GEOG_DATAROOT terrestrial-data-directory
setenv EXT_DATAROOT degribbed-files-directory
setenv DATAROOT top-level-domain-directory
setenv MOAD_DATAROOT specific-domain-directory
setenv TEMPLATES template-directory
```

The following default values will be used if these environment variables are not set prior to installation:

```
setenv SOURCE_ROOT $cwd/wrfsi_nmm
setenv INSTALLROOT $cwd/wrfsi_nmm
setenv GEOG_DATAROOT $INSTALLROOT/extdata
setenv EXT_DATAROOT $INSTALLROOT/extdata
setenv DATAROOT $INSTALLROOT/domains
setenv TEMPLATES $INSTALLROOT/templates
```

The settings for these paths at installation whether defined by environment variables or default values, can be found in file ***$INSTALLROOT/config_paths*** for later reference.

Hint: Do not use wrfsi_nmm/data/ directory for $DATAROOT.

Hint: Setting the environment variables correctly is critical every time you run the SI package. The run scripts and source code key on these environment variables to access data.

Note: The directory where the input GriB files reside is not defined through an environment variable. It is defined in the ***grib_prep*** namelist file.

- The other environment variable to set prior to installation is the location of the netCDF library:

  setenv NETCDF /path/to/netcdf-libraries

Note: Make sure the NETCDF environment variable points to ***netCDF*** libraries compiled using the same compiler used to compile the SI code.

Once the appropriate environment variables are set, run the installation script (install_wrfsi.pl) that resides in the top-directory of WRF-NMM SI package.

- To run ***"install_wrfsi.pl"*** script, first change to the ***wrfsi_nmm*** directory and then type:
  perl install_wrfsi.pl
- At the prompt inquiring whether to install the GUI, enter "***y***" to install the GUI and "***n***" to skip the GUI installation. (The GUI is a graphical user interface designed to provide tools for setting up WRF domains and an efficient method for manipulating the SI software package. For more information on the SI GUI, visit http://wrfsi.noaa.gov/)

If the installation is successful, the following executables can be found in the `$INSTALLROOT/bin` directory:

grib_prep.exe
gridgen_model.exe
hinterp.exe
siscan
vinterp.exe

The perl scripts for running the various components of the WRF-NMM SI package are in `$INSTALLROOT/etc`:

window_domain_rt.pl
grib_prep.pl
wrfprep.pl

The install script also builds the ***EXT_DATAROOT***, and ***DATAROOT*** directories according to the environment variable setting. Five subdirectories are created under ***EXT_DATAROOT***: ***extprd, static, work, log*** and ***GEOG*** (Note: The GEOG subdirectory can be ignored if **GEOG_DATAROOT** was set prior to installation).

- Running the SI for a different domain does not require reinstallation of the SI package. Rather, the user simply needs to reset the **MOAD_DATAROOT** environment variable:

  setenv MOAD_DATAROOT new-domain-directory

If the installation process included building the GUI, the executable will be ***$INSTALLROOT/wrf_tools.***

If the installation is not successful, the problem may lie with the location of the system utilities (i.e., not in the location specified in the provided ***makefile***) and/or the user's path (i.e., the location of the system utilities is not included in the user's path). To address problems with the system utilities specifications, ***cd*** to ***src/include*** directory, find the ***makefile*_machine-type*.inc.in*** that is the closest match to your machine type and edit where appropriate. Once these edits are complete, check to make sure your path includes the directories of all the relevant system utilities before restarting the installation process.

Description of the Namelist Variables

The WRF-NMM SI uses a namelist file, ***wrfsi.nl***, to specify the configuration of the SI to be used for a particular simulation or forecast. A sample namelist file can be found in the ***$TEMPLATES/default*** directory. The variables specified in the namelist are divided into six sections.

A. project_id

This section is used to fill in metadata entries that document the run. The settings in this section will not affect the WRF-NMM SI run, but are provided for convenience to allow the user to document their run in the output metadata.

1. SIMULATION_NAME: String describing the experiment or run.
2. USER_DESC: String describing who is running the configuration.

B. filetimespec

This section provides the start and stop times that bound the period for which the SI will process data. Times are in UTC

START_YEAR: 4-digit UTC year for start time.
START_MONTH: 2-digit UTC month for start time.
START_DAY: 2-digit UTC day of month for start time
START_MINUTE: 2-digit UTC minute for start time.
START_SECOND: 2-digit UTC second for start time.
END_YEAR to END_SECOND: Same as 1-5, but for ending time.
INTERVAL: Interval in seconds between output times. The output interval cannot be finer than the available data interval.

Note: When using the perl script ***wrfprep.pl*** to run ***hinterp.exe*** and ***vinterp.exe***, the user must use command line options, instead of parameters in the ***filetimespec*** section of ***wrfsi.nl***, to communicate non-real-time timing parameters to these programs (See discussion below.)

C. hgrdspec

This section is used to define the horizontal WRF-NMM grid.

NUM_DOMAINS: Integer number of nests. Nesting is not currently available for WRF-NMM. User should set to 1.

XDIM**:** Number of mass points in the X direction for an odd-numbered row (see discussion and diagram below).

YDIM: Number of rows in the Y direction (MUST BE ODD!)

Schematic of how XDIM and YDIM apply on the WRF-NMM's E-grid:

```
H V H V H V H (V)
V H V H V H V (H)
H V H V H V H (V)
V H V H V H V (H)
H V H V H V H (V)
```

In this example, H represents mass variables (e.g., temperature, pressure, moisture), while V represents vector wind quantities. The (H) and (V) at the end of the row are a so-called phantom column that is used so arrays will be completely filled (XDIM, YDIM) for both mass and wind quantities, but the phantom column does not impact the integration. In this example, XDIM=4 and YDIM=5.

PARENT_ID to DOMAIN_ORIGIN_URJ: Nesting parameters – not relevant to WRF-NMM

MAP_PROJ_NAME: Character string specifying type of map projection. The **only** valid entry for the WRF-NMM is:

"rotlat" -> Rotated Latitude and Longitude

MOAD_KNOWN_LAT/MOAD_KNOWN_LON: Real latitude and longitude of the grid's center point. Values are in degrees, with positive latitude for the northern hemisphere and negative longitude for the western hemisphere. Latitude must be between -90 and 90, and longitude between -180 and 180.

MOAD_KNOWN_LOC: String specifying the location of the known latitude and longitude. The WRF-NMM grid is always defined based on a center latitude/longitude value. Hence **MOAD_KNOWN_LOC = 'center'** should always be used for WRF-NMM.

MOAD_STAND_LATS: For the WRF-NMM, this parameter is equal to

MOAD_KNOWN_LAT. It must be between -90 and 90.

MOAD_STAND_LONS: For the WRF-NMM, this parameter is equal to

MOAD_KNOWN_LON. It must be between -180 and180.

MOAD_DELTA_X: Floating point value specifying the grid spacing in the west-east direction ***in degrees***. The spacing is in terms of the distance between a mass (H) point and its neighbor wind (V) point on the same row.

MOAD_DELTA_Y: Floating point value specifying the grid spacing in the north-south direction ***in degrees***. The spacing is in terms of the distance between a mass (H) point and its neighbor wind (V) point within the same column.

For the rotated latitude-longitude grid, the grid center is the equator. ***MOAD_DELTA_X*** and ***MOAD_DELTA_Y*** are constant within this rotated grid framework. However, in a true Earth sense, the grid spacing varies in kilometers slightly between the center latitude and the northern and southern edges due to convergence of meridians away from the equator. This behavior is more notable for domains covering a wide range of latitudes. Typically, MOAD_DELTA_X is set to be slightly larger than MOAD_DELTA_Y to counter the effect of meridional convergence, and keep the unrotated, "true earth" grid spacing more uniform over the entire grid.

The relationship between the fraction of a degree specification for the E-grid and the more typical grid spacing specified in kilometers for other grids can be approximated by considering the following schematic:

```
V -dx- H
 |       / |
dy dX dy
 | /       |
H - dx- V
```

The horizontal grid resolution is taken to be the shortest distance between two mass (H) points (diagonal – dX), while MOAD_DELTA_X (dx) and MOAD_DELTA_Y (dy) definitions refer to distances between adjacent H and V points. The distance between the H points in the diagram above is the hypotenuse of the triangle with legs dx and dy. Assuming 111 km/degree (a reasonable assumption for the rotated latitude-longitude grid) the grid spacing in km is approximately equal to: 111.0 * (SQRT (dx**2 + dy**2)).

SILAVWT_PARM_WRF - TOPTWVL_PARM_WRF: Settings for terrain smoothing within the ARW version of WRF SI. Do not apply to WRF-NMM SI.

Topography smoothing for WRF-NMM SI:

The WRF-NMM SI package currently allows a limited amount of topography smoothing. This smoothing is controlled within the code, rather then by a namelist option. The primary purpose of the subroutine ***smdhl,*** found in ***src/grid/gridgen_model.f***, is to smooth topography along the model boundaries. An additional, experimental option found in this routine smoothes localized peaks. As coded now, a five-point smoother is applied to a given point if it is higher than each of the four surrounding points, This smoothing acts to shave off the mountain tops, but does not smooth valleys out of existence. This option can be turned off by setting the parameter ***"hthresh"***, which is currently set to zero, to a large value, so a given point will never be ***"hthresh"*** higher than the four surrounding points

D. sfcfiles Section

This section is used to specify the paths to tiled global geographical data sets, which can be obtained from ftp://aftp.fsl.noaa.gov/divisions/frd-laps/WRFSI/Geog_Data EACH DATA SET MUST BE IN ITS OWN SUBDIRECTORY! Note: These paths defined in the template wrfsi.nl were set according to the GEOG_DATAROOT environment variable setting at the time of installation and do not need to be changed unless the user decides to change the location of these data sets

TOPO_30S: Path to the USGS-derived 30-second topographical height data.
LANDUSE_30S: Path to the tiled 24-category USGS 30-second land usage categorical data.
SOILTYPE_TOP_30S: Path to the FAO top-layer 16-category soil-type data.
SOILTYPE_BOT_30S: Same as SOILTYPE_TOP_30S but for bottom layer.
GREENFRAC: Path to the greenness fraction data. Resolution: 0.15 degree.
SOILTEMP_1DEG: Path to the annual mean deep-layer temperature data. Resolution: 1 degree.
ALBEDO_NCEP: Path to the monthly climatological albedo data set (normalized to local zenith). Resolution: 0.15 degree.
MAXSNOWALB: Path to climatological maximum snow albedo data. Resolution: 0.15 degree.
ISLOPE: slope data. (Not yet used by WRF). Resolution: 1 degree

E. interp_control

This section controls the horizontal and vertical interpolation of the input gridded data sets.

NUM_DOMAINS: integer number of nests, excluding the parent domain. For WRF-NMM, set to 1.

DOMAIN_ID_LIST: a list of the nests to create WRF-SI output for. For example, if you have set to configure 4 domains, then set this namelist to 2, 3, and 4.

PTOP_PA: Specifies model top in Pascals. Default is 5000 Pa. The horizontal interpolation code expects to find this topmost level within the input GRIB file, and will stop if it does not find it. The "***hinterp"*** code is revised for WRF-NMM to avoid this difficulty, but setting the model top to 5000 Pa is probably fine for most users.

HINTERP_METHOD: Integer specifying method of interpolation for atmospheric variables. Codes:

0: Nearest neighbor (not recommended)
1: 4-pt bilinear (use if input data has similar resolution as output)
2: 16-point

LSM_HINTERP_METHOD: Integer specifying the method of interpolation used for the land-masked fields. Codes are the same as above. Recommended options are: 0 or 1. ***NOTE**: **For users wishing to use the background data's landuse and soil type categories, this parameter must be set to 0 and "vegcat" and "soilcat" must be obtained from the input data set, where vegcat is a 2d array of dominant landuse category (usgs 24-cat) and soilcat is the 2d array of fao dominant soil category***

NUM_INIT_TIMES: Integer currently set to 1. This parameter controls the number of output times to use the prefix specified by "INIT_ROOT" and "LSM_ROOT". The idea is for future support of analysis "nudging". The code will use the data specified by the INIT_ROOT/LSM_ROOT for time periods 1: NUM_INIT_TIMES, and then switch to "LBC_ROOT" for the remaining time periods. Most users will set this parameter to 0, in which case all data will come from LBC_ROOT and CONSTANTS_FULL_NAME. Setting this parameter to 1 allows the model to be initialized with a different source of data for the initial conditions and land surface than what is used for the lateral boundary conditions

INIT_ROOT: Prefix of data to use for 1:NUM_INIT_TIMES. The wrfprep.pl script will look in ANALPATH (see **si_paths section**) for files with this prefix and a time string suffix valid for the desired time. This entry is only used if NUM_INIT_TIMES > 0.

LBC_ROOT: Prefix of data files to use for lateral boundary condition times. The wrfprep.pl script will link in all files in "LBCPATH" that have this prefix and a valid time suffix in the correct range.

LSM_ROOT: For each NUM_INIT_TIME (when NUM_INIT_TIMES >0), the wrfprep.pl script will link in a file with this prefix found in LSMPATH that matches in time. This option is designed to support data for the NOAH LSM coming from a source other than the INIT_ROOT.

CONSTANTS_FULL_NAME: Specifies a list of file names to look for in "CONSTANTS_PATH". Data contained in any of these files actually found will be used at every output time, and will take precedence over duplicate data found in LSM_ROOT/INIT_ROOT/LBC_ROOT files.

VERBOSE_LOG: Logical. Setting to true provides logging for troubleshooting purposes.

OUTPUT_COORD: String used to communicate to the vertical interpolation code that it is interpolating to the WRF-NMM's hybrid vertical coordinate. '***NMMH'*** should **NOT** be changed.

LEVELS: List of levels to use in the WRF modeling system in ascending order. These values range from 1.0 to 0.0. The number of levels in the list specifies the number of vertical layer interfaces. The specified values determine the distribution of the vertical layers.

OUTPUT_VARS: Controls which variables are output. ***MUST*** be set to ***'2'*** because WRF-NMM SI code expects this option.

OUTPUT_FILE_TYPE: The default is "WRF". This option generates ***netCDF*** output files from WRF-NMM SI, which is consistent with the ***/real_nmm.f/*** code provided with WRFV2.1.

F. si_paths

This section specifies the paths to the degribbed data (grib_prep output) files. In most cases, all of these parameters will be the same path (***$EXT_DATAROOT/extprd***)

ANALPATH: Path to files with INIT_ROOT prefix when NUM_INIT_TIMES > 0.

LBCPATH: Path to files with LBC_ROOT prefix for time periods >NUM_INIT_TIMES.

LSMPATH: Path to files with LSM_ROOT prefix for all time periods 0:NUM_INIT_TIMES.

CONSTANTS_PATH: Path to files with CONSTANTS_FULL_NAME for every time period.

How to Run WRF-NMM SI

Using the GUI to run the WRF-NMM SI package is recommended. If installation of the GUI was successful, type the following to start this application

$INSTALLROOT/wrf_tools

For detailed instructions on how to use the GUI, visit http://wrfsi.noaa.gov/. One of the advantages of using the GUI is that it has graphics to help locate a domain. However, going through the following steps may be helpful to understand what the various programs/scripts do and what they produce.

The following instructions are for running WRF-NMM SI manually:

NOTE: Before proceeding with Step 1 below, the user needs to obtain the required tiled geographical data sets (see discussion of wrfsi.nl sfcfiles section above) and put these files in the directory corresponding to GEOG_DATAROOT.

Set the MOAD_DATAROOT environment variable to the path where the user's domain will run. For example,

setenv MOAD_DATAROOT $INSTALLROOT/domains/my_domain

This step is not necessary if the MOAD_DATAROOT setting prior to the SI package installation is to be used.

cd to the **$TEMPLATES/** directory and copy the ***default/*** directory to a user defined domain directory:

cp -r default my-case

Remove file/directory protections:

chmod -R u+w my-domain

cd to **my-domain**/ and edit ***wrfsi.nl***.

The important ***namelist*** variables to be edited at this time are those in the ***'hgridspec'*** section as shown below (for details see "Description of the Namelist Variables" section). WRF-NMM specific parameters are shown in bold.

```
&hgridspec
NUM_DOMAINS = 1
XDIM = 123,
YDIM = 201,
PARENT_ID = 1,
RATIO_TO_PARENT = 1,
DOMAIN_ORIGIN_PARENT_X = 1,
DOMAIN_ORIGIN_PARENT_Y = 1,
MAP_PROJ_NAME = 'rotlat',
MOAD_KNOWN_LAT = 40.0,
MOAD_KNOWN_LON = -115.0,
MOAD_KNOWN_LOC = 'center'
MOAD_STAND_LATS = 40.0, 40.0,
MOAD_STAND_LONS = -115.0,
MOAD_DELTA_X = 0.0534521
MOAD_DELTA_Y = 0.0526316
SILAVWT_PARM_WRF = 0.
TOPTWVL_PARM_WRF = 2.
```

Note: MAP_PROJ_NAME = 'rotlat' and **MOAD_KNOWN_LOC = 'center'** should **NOT** be changed. **MOAD_DELTA_X** and **MOAD_DELTA_Y** **MUST** be in degrees.

Once the ***'hgridspec'*** section of the ***wrfsi.nl*** has been edited to reflect the user's domain configuration, the user is ready to run the domain localization script. A typical run command looks like:

$INSTALLROOT/etc/window_domain_rt.pl ***-w wrfsi.rotlat*** -t $TEMPLATES/*my-domain*

Other options are available for this Perl script. To obtain a listing of all options, type:

$INSTALLROOT/etc/window_domain_rt.pl -h

If the domain localization is successful, the following directories will appear under ***$MOAD_DATAROOT: static/, cdl/, siprd/, wrfprd/*** and ***log/***. Check the ***static/*** directory to make sure a file named ***static.wrfsi.rotlat*** was created. If so, the domain localization is finished. If not, check the ***localization_domain.log.date*** file in the ***log/*** directory, and try to identify errors.

The Perl script makes use of the ***wrfsi.nl*** file in the ***$TEMPLATES/my-domain/*** directory to create control files for the netCDF output in the ***cdl/*** directory and a working ***wrfsi.nl*** in the ***$MOAD_DATAROOT/static/*** directory, and then executes ***gridgen_model.exe***. The Perl script also creates a ***siprd*** directory under ***$MOAD_DATAROOT***, which will be used to store files generated by running ***wrfprep.pl*** (see Step 3).

To start over using the same MOAD_DATAROOT, add option ***'-c'*** at the end of the above run command. This option cleans the MOAD_DATAROOT directory:

***$INSTALLROOT/etc/window_domain_rt.pl -w wrfsi.rotlat -t $TEMPLATES/my-domain* -c**

Step 2: Degrib GriB files - *grib_prep.pl*

NOTE: The user is responsible for obtaining meteorological data to run WRF simulations/forecasts. Data can be obtained from a number of sites. More information on obtaining meteorological data is available at: (insert link to page describing data sources)

The degrib portion of the SI package uses the namelist file ***grib_prep.nl*** to specify the configuration information for the grib_prep program. This namelist file is in the ***$EXT_DATAROOT/static*** directory. The parameters specified in this namelist are divided into two sections*:* ***filetimespec*** and ***gpinput_defs***. If using the Perl script grib_prep.pl to run grib_prep.exe, the ***filetimespec*** section can be ignored because the time information needs to be provided via the command line (see example below). The ***gpinput_defs*** section defines the paths to the raw data, Vtable names, and filename prefixes and suffixes. This section is used by ***grib_prep.pl*** to look for the correct files matching the desired times and soft link these grib files to the work directory along with the edited namelist prior to execution of the FORTRAN program.

SRCNAME: strings (comma-separated) designating user-defined names for each type of "source" data to be processed. These names define the prefix of the final output files in ***$EXT_DATA_ROOT/extprd***. The same names must be used in the ***interp_control*** section of wrfsi.nl for ***hinterp*** to read the ***grib_prep*** output.

SRCVTAB: string (one entry per SRCNAME entry) defining the suffix used to obtain the appropriate Vtable from the ***EXT_DATA_ROOT/static*** directory to properly extract the variables from the grib files.

SRCPATH: string (one entry per SRCNAME) defining the path to each of the SRCNAME GRIB directories.

SRCCYCLE: integer (one entry per SRCNAME) defining the number of hours between new model runs of the SRCNAME model.

SRCDELAY: integer (one entry per SRCNAME) defining the number of hours after the SRCNAME's initial valid time that the entire run is expected to be available.

Note: The parameters SRCCYLE and SRCDELAY are only relevant to real-time forecasts. These parameters are used in the following manner: When grib_prep.pl runs, it uses the system clock to determine the latest available cycle that will fulfill the need for SRCNAME. For example, if the script is run at 14Z for a model that is on a 6 hour cycle with a 3 hour delay, the script will assume it is too early for a 12Z run to be available (14Z-12Z is less than the 3 hour delay), so the script will look for the 06Z cycle.

Prior to running grib_prep, edit ***grib_prep.nl*** such that the entries in ***gpinput_defs*** correspond to the type of input data to be processed and the location of this data. For instance, if the user wants to process NAM files that are located in /public/data/NAM/, ***grib_prep.nl*** should have the following entries:

```
        SRCNAME = 'NAM'
        SRCVTAB = 'NAM'
        SRCPATH = '/public/data/NAM',
```

A typical run command looks like:

$INSTALLROOT/etc/grib_prep.pl -s 2005042700 -l 12 -t 6 NAM

This run starts with 2005042700, processes data for a 12 h forecast at 6 hour interval. The data processed is from NAM, which corresponds to ***Vtable.NAM***. As shown, the time information is provided via the command line.

Hint: Use the time interval between the available data only - there is practically no advantage to interpolating data to a time interval that is smaller than provided.

Other options on the command line are available. Type the following to see all options:

> ***grib_prep.pl -h***

The output from running this script should reside in the ***`$EXT_DATAROOT/extprd/`*** directory, with file names beginning with NAM:

> ***NAM:2005-04-27_00***
> ***NAM:2005-04-27_06***
> ***NAM:2005-04-27_12***

If these files are not created, check ***$EXT_DATAROOT/log/gp_NAM.2005042700.log*** to find clues.

Hint: **grib_prep.exe** runs best when only the files relevant to a particular run are placed in the directory defined in ***SRCPATH***.

Step 3: Interpolating meteorological data - *wrfprep.pl*

Once the domain has been configured and the input data degribbed, the user is ready for the final step in the SI process: interpolating meteorological data to the WRF-NMM grid. The ***wrfprep.pl*** script executes both ***hinterp*** and ***vinterp*** executables.

Prior to running wrfprep.pl, edit the ***`'interp_control'`*** section of the **wrfsi.nl** file in **$MOAD_DATAROOT/static** to reflect the desired configuration:

```
&interp_control
 NUM_DOMAINS = 1,
 DOMAIN_ID_LIST = 1,
 PTOP_PA = 5000,
 HINTERP_METHOD = 1,
 LSM_HINTERP_METHOD = 0,
 NUM_INIT_TIMES = 1,
 INIT_ROOT = 'NAM',
 LBC_ROOT = 'NAM',
 LSM_ROOT = 'NAM',
 CONSTANTS_FULL_NAME = 'SSTDATA','SNOWDATA',
 VERBOSE_LOG = .true.,
 OUTPUT_COORD = 'NMMH',
 LEVELS = 1.0000, .9955, .9908, .986, .9811, .9761, .971, .9658, .9605,
 .955, .9493, .9434, .9372, .9307, .9239, .9167, .9087, .8998, .8891,
 .8771, .8636, .8494, .8342, .818, .8013, .7839, .766, .7474, .7283,
 .7086, .6884, .6675, .6459, .6237, .6008, .5775, .5539, .5300, .5060,
 .4819, .4577, .4334, .4089, .3843, .3598, .3355, .3116, .2881, .2661,
 .2453, .2261, .2081, .1899, .1706 .1481, .1238, .0992, .0744, .0496,
 .0248, .0000,
OUTPUT_VARS = 2,
OUTPUT_FILE_TYPE = 'WRF',
```

WRF-NMM specific parameters are once again shown in bold. Reminder: **`OUTPUT_COORD MUST`** be set to **`'NMMH'`** for WRF-NMM.

Make sure `INIT_ROOT`, `LBC_ROOT` and/or `LSM_ROOT` are set to the correct data type (i.e., the data type specified when running ***grib_prep.pl***). The different root names allow one to use data from different sources.

To run the interpolation script, type the following:

$INSTALLROOT/etc/wrfprep.pl -s 2050042700 -f 12

Note the time information is provided on the command line, which is required when using ***wrfprep.pl*** to run ***hinterp*** and ***vinterp*** for non-real-time cases. Again other options may be found by typing:

$INSTALLROOT/etc/wrfprep.pl -h

If successful, the following files will reside in the ***$MOAD_DATAROOT/siprd/*** directory:

hinterp.d01.2005-04-27_00:00:00
hinterp.d01.2005-04-27_06:00:00
hinterp.d01.2005-04-27_12:00:00
hinterp.global.metadata
wrf_real_input_nm.d01.2005-04-27_00:00:00
wrf_real_input_nm.d01.2005-04-27_06:00:00
wrf_real_input_nm.d01.2005-04-27_12:00:00
wrf_real_input_nm.d01.cdl

The ***wrf_real_input_nm.d0.**** files are the input files for the ***WRF-NMM real*** (***rea_nmml.exe***) program. These files are in netCDF format.

If the interpolation process is not successful, check the files in ***$MOAD_DATAROOT/log/*** for possible errors:

2005042700.wrfprep
2005042700.hinterp
2005042700.vinterp

WRF-NMM SI GUI

For a detailed description and instruction on how to use it, please visit:
http://wrfsi.noaa.gov/

If you have successfully installed the GUI, type the following to start it:

$INSTALLROOT/wrf_tools

Using Multiple Data Sources

For some applications, the user may want to use data from different sources to generate the input fields for a WRF simulation or forecast. For instance, the user may want to use sea-surface temperature (SST) from a data source that is not the same as that for the other meteorological fields, or the user may want to use fields for land-surface model that are from a different data source. The SI package supports this function. To do so, ***grib_prep.pl*** needs to be run multiple times - once for each data source. Since the special input fields are typically only needed for the model initial time, ***grib_prep*** generally only needs to process the special data sets for a single time period. The command for processing a single time period is:

$INSTALLROOT/etc/grib_prep.pl -s 2005042700 -l 0 SST

In this example, the source for the SST data is generically declared as SST, which would require the presence of Vtable.SST. Similarly, if the LSM fields come from another source, such as AGRMET, one would issue the command:

$INSTALLROOT/etc/grib_prep.pl -s 20050424700 -l 0 AGRMET

When using a different data source for the LSM fields, the user may want to remove the LSM fields from the Vtable used to obtain the other meteorological fields. Once the degribbing process is complete for all data sources, edit ***wrfsi.nl*** so ***wrfprep.pl*** knows how to use the multiple sources of input data.

Different source data can be introduced by two pathways. The first pathway is through the namelist variable 'CONSTANT_FULL_NAME'. This approach usually works for a single field like SST. Edit ***wrfsi.nl*** so that 'CONSTANT_FULL_NAME' is set to:

CONSTANT_FULL_NAME = 'SSTDATA',

The Perl script will link the SST file produced during the degrib process to SSTDATA.

To use a different data source for LSM fields, edit ***wrfsi.nl*** so that the prefix variables are set to the following:

INIT_ROOT = 'NAM',

LBC_ROOT* = *'NAM',
LSM_ROOT* = *'AGRMET',

A special example of multiple input data sources is the NCEP/NCAR Reanalysis Project (NNRP) data. Although all the fields are from the same data source, the data are archived such that the upper air and surface data have different data dimensions. To use both surface and upper air data from this source, grib_prep.pl must be run twice: once for the surface data using Vtable.NNRPSFC, and once for the upper air data using Vtable.NNRP (both provided). The following settings should then be used in the wrfsi.nl:

INIT_ROOT = 'NNRP',
LBC_ROOT = 'NNRP',
LSM_ROOT = 'NNRPSFC',

Checking WRF-NMM SI Output

WRF-NMM SI output may be checked in several ways. If all the anticipated files are created, then chances are good that everything has gone well. However, if a sanity check is desired, here are a few ways to go about it. These options include graphics tools and print outs

Checking output from *grib_prep*

Use the utility program, ***plotfmt.exe***, described above to make plots of the degribbed files. This utility is especially useful when ingesting data from a source other than those supported by the program.

The static fields created by ***gridgen_model*** can be checked by using the netCDF utility, ncdump. To obtain information (header information) on what fields are in the file created by ***gridgen_model***, type:

ncdump -h static.wrfsi.rotlat

To determine the actual values for a particular variable, type

ncdump -v variable-name static.wrfsi.rotlat

Checking output from graphics provided by GUI:

The user can display the results of SI domain localization (images of the static land characteristic files). Documentation for how to display graphics can be found at: http://wrfsi.noaa.gov/gui/users_guide/V2/users_guide.html.

Checking output from *hinterp*

Use utility program ***siscan*** described above.

Checking output from *vinterp*

By this stage, the output files are in WRF I/O API-conforming netCDF format, so various supported post-processing tools may be used. See again Chapter 7 for more information.

List of Fields in WRF-NMM SI Output

A List of Fields

```
netcdf wrf_real_input_nm.d01 {

dimensions:
        Time = UNLIMITED ; // (1 currently)
        DateStrLen = 19 ;
        west_east = 55 ;
        south_north = 91 ;
        bottom_top = 45 ;
        bottom_top_stag = 46 ;
        ext_scalar = 1 ;
        land_cat = 24 ;
        soil_cat = 16 ;
variables:
        char Times(Time, DateStrLen) ;
        float T(Time, bottom_top, south_north, west_east) ;
        float Q(Time, bottom_top, south_north, west_east) ;
        float U(Time, bottom_top, south_north, west_east) ;
        float V(Time, bottom_top, south_north, west_east) ;
        float PD(Time, south_north, west_east) ;
        float DETA(Time, bottom_top) ;
        float AETA(Time, bottom_top) ;
        float ETAX(Time, bottom_top_stag) ;
        float DFL(Time, bottom_top_stag) ;
        float DETA1(Time, bottom_top) ; WRFSI/vinterp" ;
        float AETA1(Time, bottom_top) ;
        float ETA1(Time, bottom_top_stag) ;
        float DETA2(Time, bottom_top) ;
        float AETA2(Time, bottom_top) ;
        float ETA2(Time, bottom_top_stag) ;
        float PDTOP(Time, ext_scalar) ;
        float PT(Time, ext_scalar) ;
        float PMSL(Time, south_north, west_east) ;
        float WEASD(Time, south_north, west_east) ;
        float TSK(Time, south_north, west_east) ;
        float ST000010(Time, south_north, west_east) ;
        float ST010040(Time, south_north, west_east) ;
        float ST040100(Time, south_north, west_east) ;
        float ST100200(Time, south_north, west_east) ;
        float TGROUND(Time, south_north, west_east) ;
        float SM000010(Time, south_north, west_east) ;
        float SM010040(Time, south_north, west_east) ;
        float SM040100(Time, south_north, west_east) ;
        float SM100200(Time, south_north, west_east) ;
        float XICE(Time, south_north, west_east) ;
```

```
          float SST(Time, south_north, west_east) ;
          float CANWAT(Time, south_north, west_east) ;
          float SPECHUMD(Time, south_north, west_east) ;
          float SOILHGT(Time, south_north, west_east) ;
          float GLAT(Time, south_north, west_east) ;
          float GLON(Time, south_north, west_east) ;
          float LAT_V(Time, south_north, west_east) ;
          float LON_V(Time, south_north, west_east) ;
          float LANDMASK(Time, south_north, west_east) ;
          float SM(Time, south_north, west_east) ;
          float Z0(Time, south_north, west_east) ;
          float FIS(Time, south_north, west_east) ;
          float VEGFRC(Time, south_north, west_east) ;
          float GREENMAX(Time, south_north, west_east) ;
          float GREENMIN(Time, south_north, west_east) ;
          float ALBASE(Time, south_north, west_east) ;
          float MXSNAL(Time, south_north, west_east) ;
          float SOILTB(Time, south_north, west_east) ;
          float ISLOPE(Time, south_north, west_east) ;
          float CMC(Time, south_north, west_east) ;
          float LANDUSEF(Time, land_cat, south_north, west_east) ;
          float SOILCTOP(Time, soil_cat, south_north, west_east) ;
        float SOILCBOT(Time, soil_cat, south_north, west_east) ;
         float ZNW(Time, bottom_top_stag) ;
         float MU0(Time, south_north, west_east) ;
         float T(Time, bottom_top, south_north, west_east) ;
         float QVAPOR(Time, bottom_top, south_north, west_east) ;
         float U(Time, bottom_top, south_north, west_east_stag) ;
         float V(Time, bottom_top, south_north_stag, west_east) ;
         float SPECHUMD(Time, bottom_top, south_north, west_east) ;
         float PMSL(Time, south_north, west_east) ;
         float SNOW(Time, south_north, west_east) ;
         float TSK(Time, south_north, west_east) ;
         float ST000010(Time, south_north, west_east) ;
         float ST010040(Time, south_north, west_east) ;
         float ST040100(Time, south_north, west_east) ;
         float ST100200(Time, south_north, west_east) ;
         float SM000010(Time, south_north, west_east) ;
         float SM010040(Time, south_north, west_east) ;
         float SM040100(Time, south_north, west_east) ;
         float SM100200(Time, south_north, west_east) ;
         float XICE(Time, south_north, west_east) ;
         float CANWAT(Time, south_north, west_east) ;
         float SOILHGT(Time, south_north, west_east) ;
         float XLAT(Time, south_north, west_east) ;
         float XLONG(Time, south_north, west_east) ;
         float LANDMASK(Time, south_north, west_east) ;
         float HGT(Time, south_north, west_east) ;
         float TOPOSTDV(Time, south_north, west_east) ;
         float TOPOSLPX(Time, south_north, west_east) ;
         float TOPOSLPY(Time, south_north, west_east) ;
         float COSALPHA(Time, south_north, west_east) ;
         float SINALPHA(Time, south_north, west_east) ;
         float F(Time, south_north, west_east) ;
         float E(Time, south_north, west_east) ;
         float MAPFAC_M(Time, south_north, west_east) ;
         float MAPFAC_U(Time, south_north, west_east_stag) ;
```

```
 float MAPFAC_V(Time, south_north_stag, west_east) ;
 float VEGFRA(Time, south_north, west_east) ;
 float SHDMAX(Time, south_north, west_east) ;
 float SHDMIN(Time, south_north, west_east) ;
 float ALBBCK(Time, south_north, west_east) ;
 float SNOALB(Time, south_north, west_east) ;
 float TMN(Time, south_north, west_east) ;
 float SLOPECAT(Time, south_north, west_east) ;
 float LANDUSEF(Time, land_cat, south_north, west_east) ;
 float SOILCTOP(Time, soil_cat, south_north, west_east) ;
float SOILCBOT(Time, soil_cat, south_north, west_east) ;
```

Global Attributes in WRF-NMM SI Output File

```
:FLAG_ST000010 = 1 ;
   :FLAG_ST010040 = 1 ;
   :FLAG_ST040100 = 1 ;
   :FLAG_ST100200 = 1 ;
   :FLAG_SM000010 = 1 ;
   :FLAG_SM010040 = 1 ;
   :FLAG_SM040100 = 1 ;
   :FLAG_SM100200 = 1 ;
   :FLAG_SST = 1 ;
   :FLAG_TOPOSOIL = 1 ;
   :simulation_name = "WRF-NMM using NAM" ;
   :user_desc = "NCAR/RAL/DTC" ;
   :si_version = 2 ;
   :anal_version = 0 ;
   :wrf_version = 0 ;
   :post_version = 0 ;
   :map_projection = "ROTATED LATLON" ;
   :moad_known_lat = 42.f ;
   :moad_known_lon = -115.f ;
   :moad_known_loc = "CENTER" ;
   :moad_stand_lats = 40.f, 0.f, 0.f, 0.f ;
   :moad_stand_lons = -71.f, 0.f, 0.f, 0.f ;
   :moad_delta_x = 14714.16f ;
   :moad_delta_y = 14714.16f ;
   :horiz_stagger_type = "AR-E" ;
   :num_stagger_xy = 2 ;
   :stagger_dir_x = 0.f, 0.f, 0.f, 0.f ;
   :stagger_dir_y = 0.f, 0.f, 0.f, 0.f ;
   :num_stagger_z = 1 ;
   :stagger_dir_z = 0.f, 0.f ;
   :vertical_coord = "NMMH" ;
   :num_domains = 1 ;
   :init_date = 20050123 ;
   :init_time = 0.f ;
   :end_date = 20050124 ;
   :end_time = 0.f ;
   :lu_source = "USGS" ;
   :lu_water = 16 ;
   :lu_ice = 24 ;
   :st_water = 14 ;
   :TITLE = "OUTPUT FROM WRF SI V02 PREPROCESSOR" ;
```

```
:START_DATE = "2005-01-23_00:00:00.0000" ;
:CEN_LAT = 42.f ;
:CEN_LON = -71.f ;
:TRUELAT1 = 42.f ;
:TRUELAT2 = 42.f ;
:MAP_PROJ = 203 ;
:DYN_OPT = 4 ;
:ISWATER = 16 ;
:ISICE = 24 ;
:MMINLU = "USGS" ;
:WEST-EAST_GRID_DIMENSION = 55 ;
:SOUTH-NORTH_GRID_DIMENSION = 91 ;
:BOTTOM-TOP_GRID_DIMENSION = 40 ;
:DX = 14714.16f ;
:DY = 14714.16f ;
:P_TOP = 5000.f ;
```